

# XAMARIN ANDROID : Bekerja Dengan CardView

## ***Lisensi Dokumen:***

*Copyright © 2003 IlmuKomputer.Com*

*Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.*

**Junindar, ST, MCPD, MOS, MCT, MVP**

*junindar@gmail.com*

<http://junindar.blogspot.com>

## Abstrak

Android adalah salah satu *mobile oprating system* yang saat banyak digunakan di berbagai *smartphone*, baik yang berspesifikasi rendah maupun mumpuni. Dikarenakan android saat ini merupakan salah satu mobile OS paling digemari, maka tidak dipungkiri sangat banyak sekali aplikasi-aplikasi mobile untuk android. Untuk membuat aplikasi mobile ini ada beberapa tools yang bisa digunakan dan salah satunya adalah dengan menggunakan Xamarin.

## Pendahuluan

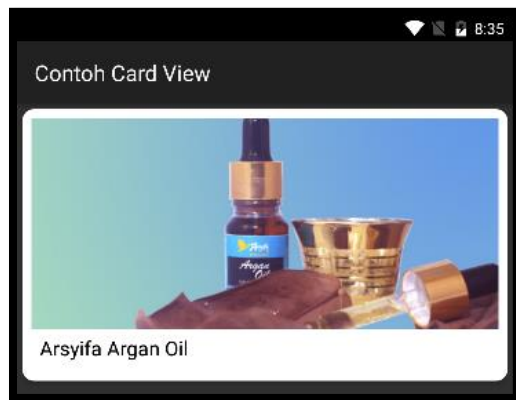
Sebelum memulai latihan pada artikel ini, disarankan untuk terlebih dahulu membaca dan membuat latihan pada artikel sebelumnya yang bisa didapatkan disini <http://junindar.blogspot.com/2018/04/xamarin-android-bekerja-dengan-listview.html> dan <http://junindar.blogspot.com/2019/01/xamarin-android-bekerja-dengan.html>.

Pada artikel ini akan dijelaskan bagaimana bekerja dengan CardView pada aplikasi android dengan menggunakan Xamarin.

CardView adalah widget yang tersedia pada Android 5.0 (Lollipop), dimana widget ini menampilkan item atau konten dalam berbentuk card (kartu). Sedangkan pada konten CardView bisa terdapat text dan image. CardView merupakan turunan dari FrameLayout widget, sedangkan bentuk atau ciri-cirinya terdapat bayangan dan sudut bulat (rounded corner). Untuk menggunakan CardView kita harus menginstall 3 buah Nuget dari xamarin yaitu `Android.Support.v7.AppCompat`, `Android.Support.v4` dan `Android.Support.v7.CardView`. Yang perlu diketahui properti-properti untuk mengatur tampilan widget.

- `cardCornerRadius` : Untuk mengatur radius sudut pada layout.
- `cardBackgroundColor` : Untuk mengatur warna Background.
- `cardElevation` : Untuk mengatur bayangan.

Tools yang kita gunakan untuk membangun aplikasi ini adalah Xamarin. Xamarin adalah salah tools untuk membuat aplikasi mobile dimana bahasa pemograman yang digunakan adalah C#. Xamarin merupakan perusahaan perangkat lunak yang berdiri pada bulan Mei tahun 2011. *Integrated development environment* (IDE) pada Xamarin di kenal dengan nama Xamarin Studio. Tetapi selain Xamarin Studio kita juga dapat menggunakan Visual Studio dan pada artikel ini kita akan menggunakan Visual Studio 2015.



## Isi

CardView adalah widget yang tersedia pada Android 5.0 (Lollipop), dimana widget ini menampilkan item atau konten dalam berbentuk card (kartu). Sedangkan pada konten CardView bisa terdapat text dan image. CardView merupakan turunan dari FrameLayout widget, sedangkan bentuk atau ciri-cirinya terdapat bayangan dan sudut bulat (rounded corner). Untuk menggunakan CardView kita harus menginstall 3 buah Nuget dari xamarin yaitu `Android.Support.v7.AppCompat`, `Android.Support.v4` dan `Android.Support.v7.CardView`. Yang perlu diketahui properti-properti untuk mengatur tampilan widget.

- `cardCornerRadius` : Untuk mengatur radius sudut pada layout.
- `cardBackgroundColor` : Untuk mengatur warna Background.
- `cardElevation` : Untuk mengatur bayangan.

Untuk membahas CardView, kita akan membuat dua latihan yang pertama menampilkan satu baris data menggunakan CardView, dan yang kedua menampilkan data multi row dengan menggunakan ListView dan CardView.

Sebelumnya buat sebuah layout untuk CardView dengan nama "CardViewLayout.xml".

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:cardview="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center_horizontal"
    android:padding="5dp">
```

```
<android.support.v7.widget.CardView
    android:layout_width="fill_parent"
    android:layout_height="245dp"
    android:layout_gravity="center_horizontal"
    cardview:cardElevation="4dp"
    cardview:cardBackgroundColor="#ffffff"
    cardview:cardCornerRadius="10dp">
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="240dp"
        android:orientation="vertical"
        android:padding="8dp">
        <ImageView
            android:layout_width="fill_parent"
            android:layout_height="190dp"
            android:id="@+id/imgView"
            android:scaleType="centerCrop" />
        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="5dp"
            android:textAppearance="?android:attr/textAppearanceMedium"
            android:textColor="#000000"
            android:id="@+id/lblTitle"
            android:layout_gravity="center_horizontal"
            android:layout_marginLeft="5dp" />
    </LinearLayout>
</android.support.v7.widget.CardView>
</LinearLayout>
```

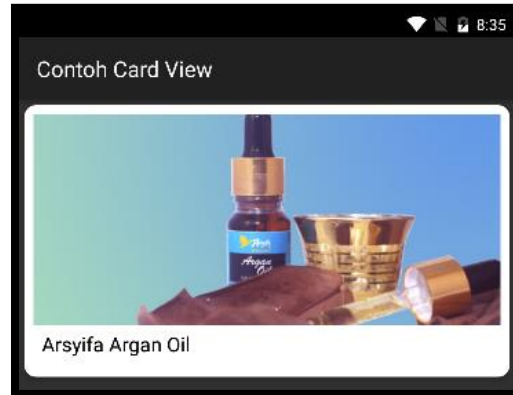
Pada sintaks xml diatas dapat dilihat untuk CardView menggunakan android.support.v7.widget.CardView. Dan didalam CardView terdapat dua widget ImageView (imgView) dan TextView (lblTitle).

- Lalu tambahkan sebuah activity dengan nama "CardViewStandardActivity.cs". Pada method OnCreate() ketikkan sintak dibawah ini.

```
SetContentView(Resource.Layout.CardViewLayout);
var lblTitle = FindViewById<TextView>(Resource.Id.lblTitle);
var imgView = FindViewById<ImageView>(Resource.Id.imgView);
lblTitle.Text = "Arsyifa Argan Oil";
imgView.SetImageResource(Resource.Drawable.gambar_arsyifa);
```

Untuk TextView akan diisi text yang kita hardcode pada Activity, sedangkan ImageView menggunakan image yang ada pada folder Drawable.

Jalankan program untuk melihat hasilnya



Setelah berhasil membuat single CardView seperti diatas, langkah selanjutnya kita akan membuat tampilan multiple CardView dengan menggunakan ListView. Seperti yang telah disebutkan untuk menampilkan multiple row atau baris kita gunakan ListView. Dan CardView berfungsi untuk menampilkan gambar dan text pada setiap item didalam ListView.

Pada ListView kita gunakan Adapter untuk menampilkan data pada screen. Jadi yang diperlukan untuk membuat latihan yang kedua ini adalah layout untuk ListView, layout CardView (telah kita buat sebelumnya), dan custom Adapter. Pertama-tama untuk memulai latih CardView yang kedua ini, buat sebuah layout untuk ListView dengan nama "CardViewCustomLayout.axml". Dimana pada layout hanya terdapat sebuah widget ListView dengan id nya "makananListView".

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:local="http://schemas.android.com/apk/res-auto"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <ListView
        android:id="@+id/makananListView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</LinearLayout>
```

Langkah kedua adalah membuat custom Adapter. Pada penjelasan mengenai ListView telah dijelaskan bagaimana cara kerja dan membuat custom adapter dan digunakan oleh ListView untuk menampilkan datanya. Seperti pada latihan ListView disini kita juga menggunakan class Makanan yang telah dibuat sebelumnya.

Sedangkan untuk custom adapter, buat sebuah class dengan nama “MakananCardAdapter”, dimana isinya hampir sama dengan MakananAdapter dan hanya beberapa bagian yang mengalami perubahan. Untuk perubahan sintaks akan dijelaskan dibawah ini.

Note : *Masukkan beberapa gambar pada folder Drawable sesuai dengan jumlah data pada class MakananRepository.*

```
public class MakananCardAdapter : BaseAdapter<Makanan>
{
    private readonly Activity _context;
    private readonly List<Makanan> _listMakanan;
    private readonly List<int> _imageResId;

    public MakananCardAdapter(Activity context, List<Makanan>
listmakanan)
    {
        _context = context;
        _listMakanan = listmakanan;
        _imageResId =new List<int> {
            Resource.Drawable.nasi_goreng,
            Resource.Drawable.ketang_goreng,
            Resource.Drawable.sop_ayam,
            Resource.Drawable.ayam_bakar,
            Resource.Drawable.nasi_uduk,
            Resource.Drawable.pecel_lele};
    }
}
```

Jika kita bandingkan dengan class MakananAdapter pada ListView yang sebelumnya, tidak terdapat variable \_imageResId. Sedangkan pada constructor variable tersebut kita isikan dengan nilai integer dari ID image pada Drawable. Sedangkan pada method GetView(), kita buat sintaks seperti dibawah ini.

```
var view = convertView ?? _context.LayoutInflater.Inflate
(Resource.Layout.CardViewLayout, parent, false);
var titleTextView = view.FindViewById<TextView>(Resource.Id.lblTitle);
var imageView = view.FindViewById<ImageView>(Resource.Id.imgView);
titleTextView.Text = _listMakanan[position].NamaMakanan + " - Harga : " +
    _listMakanan[position].Harga.ToString("C");
imageView.SetImageResource(_imageResId[position]);
return view;
```

Untuk layout, kita gunakan CardViewLayout, dimana layout ini yang akan mengisi disetiap item pada ListView. Untuk TextView diisi informasi Nama Makanan dan Harga, dengan menggunakan position sebagai indexnya. Dan terakhir kita pada ImageView kita gunakan SetImageResource dengan nilai dari variable \_imageResId menggunakan position sebagai indexnya.

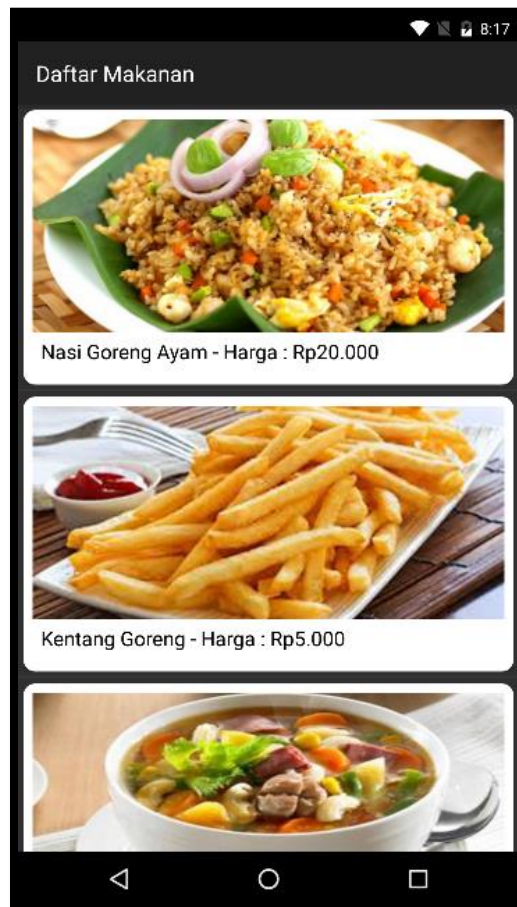
Lalu pada Activity, komen terlebih dahulu sintaks-sintaks pada method OnCreate(), lalu ganti dengan sintaks dibawah ini.

```
base.OnCreate(savedInstanceState);  
SetContentView(Resource.Layout.CardViewCustomLayout);  
var makananListView = FindViewById<ListView>(Resource.Id.makananListView);  
makananListView.ItemClick += makananListView_ItemClick;  
makananListView.FastScrollEnabled = true;  
var moviesAdapter = new MakananCardAdapter(this, MakananRepository.ListMakanan);  
makananListView.Adapter = moviesAdapter;
```

Pada SetContentView, kita ganti dengan menggunakan CardViewCustomLayout.

Selanjutnya sintaks yang lain sama seperti pada latihan ListView.

Jalankan program untuk melihat hasilnya, pastikan hasil yang didapat seperti pada gambar dibawah ini.



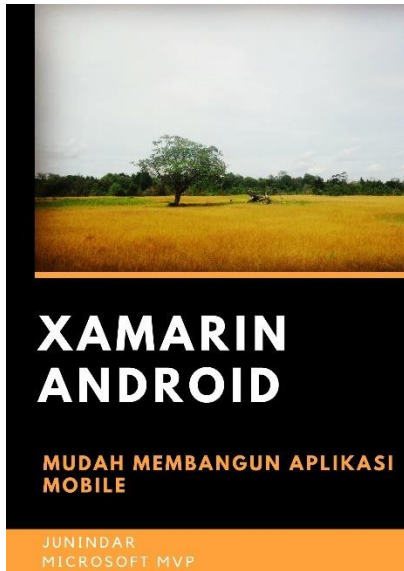


## **Penutup**

Sedangkan untuk memudahkan dalam memahami isi artikel, maka penulis juga menyertakan dengan full source code project latihan ini, dan dapat di download disini

<http://junindar.blogspot.com/2019/02/xamarin-android-bekerja-dengan-cardview.html>

## Referensi



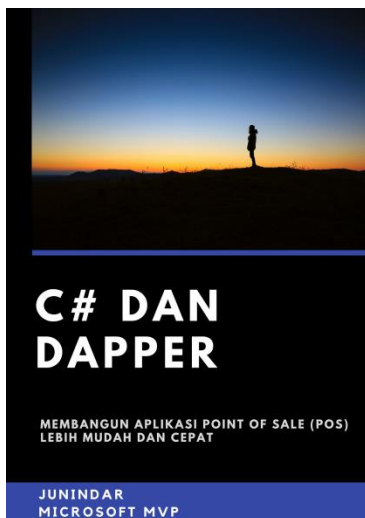
<https://play.google.com/store/books/details?id=G4tFDgAAQBAJ>



<https://play.google.com/store/books/details?id=VSLiDQAAQBAJ>



[https://play.google.com/store/books/details/Junindar\\_Xamarin\\_Forms?id=6Wg-DwAAQBAJ](https://play.google.com/store/books/details/Junindar_Xamarin_Forms?id=6Wg-DwAAQBAJ)



[https://play.google.com/store/books/details/Junindar\\_C dan Dapper Membangun Aplikasi POS P  
oint?id=6TErDwAAQBAJ](https://play.google.com/store/books/details/Junindar_C_dan_Dapper_Membangun_Aplikasi_POS_Point?id=6TErDwAAQBAJ)



[https://play.google.com/store/books/details/Junindar\\_ASP\\_NET\\_MVC\\_Membangun\\_Aplikasi\\_Web\\_Lebih?id=XLlyDwAAQBAJ](https://play.google.com/store/books/details/Junindar_ASP_NET_MVC_Membangun_Aplikasi_Web_Lebih?id=XLlyDwAAQBAJ)

## Biografi Penulis.



Junindar Lahir di Tanjung Pinang, 21 Juni 1982. Menyelesaikan Program S1 pada jurusan Teknik Inscreenatika di Sekolah Tinggi Sains dan Teknologi Indonesia (ST-INTEN-Bandung). Junindar mendapatkan Award Microsoft MVP VB pertanggal 1 oktober 2009 hingga saat ini. Senang mengutak-atik computer yang berkaitan dengan bahasa pemrograman. Keahlian, sedikit mengerti beberapa bahasa pemrograman seperti : VB.Net, C#, SharePoint, ASP.NET, VBA. Reporting: Crystal Report dan Report Builder. Database: MS Access, MY SQL dan SQL Server. Simulation / Modeling Packages: Visio Enterprise, Rational Rose dan Power Designer. Dan senang bermain gitar, karena untuk bisa menjadi pemain gitar dan seorang programmer sama-sama membutuhkan seni. Pada saat ini bekerja di salah satu Perusahaan Consulting dan Project Management di Malaysia sebagai Senior Consultant. Memiliki beberapa sertifikasi dari Microsoft yaitu Microsoft Certified Professional Developer (MCPD – SharePoint 2010), MOS (Microsoft Office Specialist) dan MCT (Microsoft Certified Trainer) Mempunyai moto hidup: **“Jauh lebih baik menjadi Orang Bodoh yang giat belajar, dari pada orang Pintar yang tidak pernah mengimplementasikan ilmunya”**.